《Python语言程序设计》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 马昆 | 班级 | 2206831 | | | 学号 | 2206831544 |
| 实验名称 | 实验六 Python面向对象程序设计 | | | | | | |
| 实验时间 | 2023年 06 月 04 日 | | | 成绩 |  | | |
| **实验目的：**  1、熟悉类的相关术语及实现方法  2、熟悉类变量的私有化类属性  3、掌握类的继承  4、了解类相关的常用内置函数和重载运算符  **实验环境：**  1、硬件环境：联想拯救者Y7000P-2022。  2、软件环境：  1)软件：Anaconda,spyder  2)操作系统：Windows 11 专业工作站版  **实验内容：**  编程练习：  1、重写实验五的练习1，要求定义一个PasswordTool类，将密码设置的相关操作封装到类中。  2、重写实验五的练习2，要求定义一个FileTool类，将文件操作封装到类中。  **实验过程及结果记录：**  编程练习：  1、重写实验五的练习1，要求定义一个PasswordTool类，将密码设置的相关操作封装到类中。   |  | | --- | | 代码  """  把文件中保存的密码读取出来  重写实验五的练习2，要求定义一个FileTool类，将文件操作封装到类中  """  # 密码工具类  class PasswordTool:  # 构造方法  def \_\_init\_\_(self, value):  self.value = value  self.strong = "000"  # 判断密码长度  if (len(value) >= 8):  self.strong = replace\_char(self.strong, "1", 0)  # 判断密码是否包含数字  if (has\_number(value)):  self.strong = replace\_char(self.strong, "1", 1)  # 判断密码是否包含字母  if (has\_letter(value)):  self.strong = replace\_char(self.strong, "1", 2)  # 转字符串  def \_\_str\_\_(self):  return "密码：%s，强度：%s\n" % (self.value, self.strong)  # 文件读写类  class FileTool:  # 构造方法  def \_\_init\_\_(self, path):  self.path = path    # 追加写入文件  def append(self, line):  f = open(self.path, "a")  f.write(line)  f.close()    # 读取文件  def read\_lines(self):  f = open(self.path, "r")  lines = f.readlines()  f.close()  return lines    # 字符串按索引位置替换字符  def replace\_char(old\_string, char, index):  return f'{old\_string[:index]}{char}{old\_string[index+1:]}'  # 判断字符串是否包含数字  def has\_number(s):  for c in s:  if ord(c) >= 48 and ord(c) <= 57:  return True  return False  # 判断字符串是否包含字母  def has\_letter(s):  for c in s:  if (ord(c) >= 65 and ord(c) <= 90) or (ord(c) >= 97 and ord(c) <= 122):  return True  return False  # 主程序开始  count = 0;  while True:  # 打印分割线  print("--------------------------")  # 不合格次数超过5次就限制输入  if (count >= 5):  print("不合格次数超过5次，禁止输入")  break    password = PasswordTool(input("请输入你的密码"))  if password.strong == "111":  count = 0  else:  count += 1  print("密码强度不合格")    # 将密码存入文件中  fileTool = FileTool("password.txt")  fileTool.append(str(password)) | | 运行结果 |   2、重写实验五的练习2，要求定义一个FileTool类，将文件操作封装到类中。   |  | | --- | | # 文件读写类  # 文件读写类  class FileTool:  # 构造方法  def \_\_init\_\_(self, path):  self.path = path    # 追加写入文件  def append(self, line):  f = open(self.path, "a")  f.write(line)  f.close()    # 读取文件  def read\_lines(self):  f = open(self.path, "r")  lines = f.readlines()  f.close()  return lines  fileTool = FileTool("password.txt")  lines = fileTool.read\_lines()  for line in lines:  print(line) | | 运行结果 | | | | | | | | |
| **实验总结：**  通过本次实验，熟悉了Python中类的相关术语以及实现方法，熟悉了类变量与成员变量，以及它们的访问权限处理。了解了类的继承，了解了类的常用内置函数。明白了面向对象的基本思想。 | | | | | | | |
|  | | | | | | | |